

# Web Information Technology Project

Academic Year 2023-2024

## Track 1: Image Gallery - Pure HTML Version

Luca De Nicola, Filippo Raimondi  
Personal Code: 10808901  
Personal Code: 10809051

## Contents

<b>1</b>	<b>Database Design</b>	<b>2</b>
1.1	ER Diagram . . . . .	2
1.2	Database Schema . . . . .	2
<b>2</b>	<b>Components</b>	<b>4</b>
2.1	Model . . . . .	4
2.2	Utils . . . . .	4
2.3	DAOs . . . . .	4
2.4	Controllors . . . . .	5
2.5	Views . . . . .	6
<b>3</b>	<b>IFML Diagram</b>	<b>6</b>
<b>4</b>	<b>Sequence Diagram</b>	<b>7</b>
4.1	Event: Sign In . . . . .	7
4.2	Event: Sign Up . . . . .	7
4.3	Event: Create Album . . . . .	7
4.4	Event: Render Image Page . . . . .	8
4.5	Event: Logout . . . . .	8
4.6	Event: Add Comment . . . . .	9
4.7	Event: Add Image . . . . .	9
4.8	Event: Delete Image . . . . .	10
4.9	Event: Render Album Page . . . . .	10

# 1 Database Design

## 1.1 ER Diagram

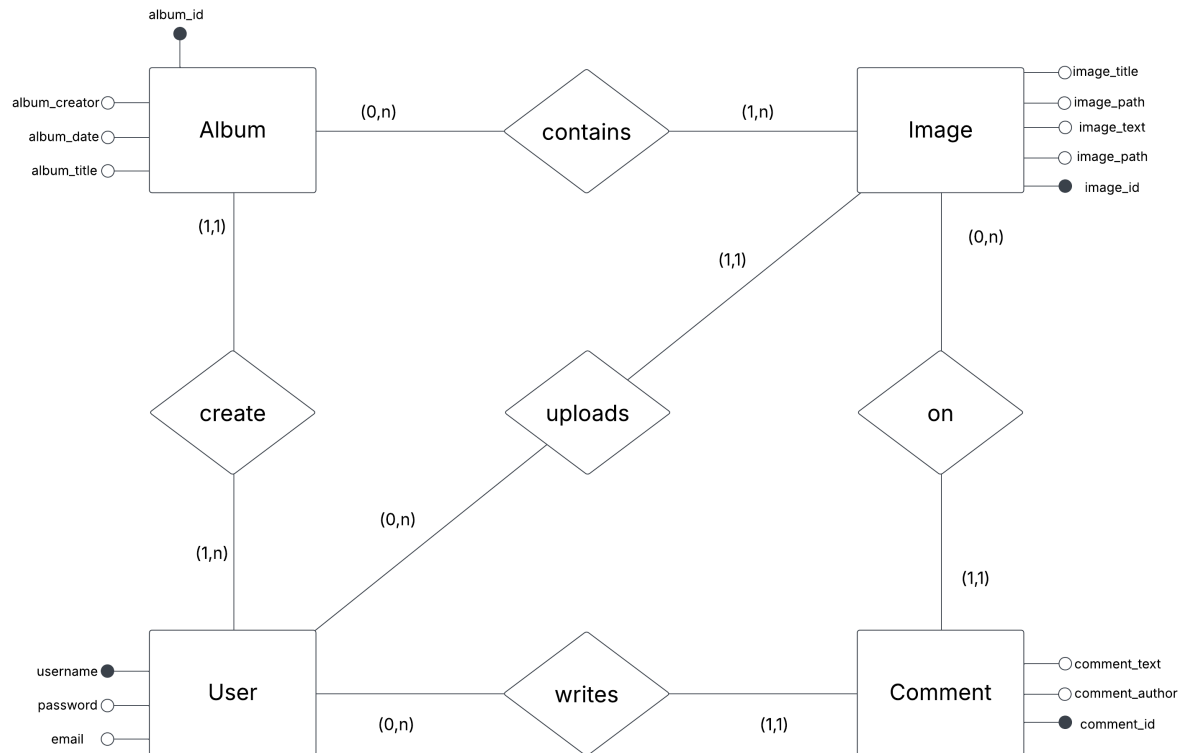


Figure 1: ER Diagram of the Database

## 1.2 Database Schema

```
1 CREATE TABLE 'User' (  
2   'username' varchar(32) NOT NULL,  
3   'email' varchar(64) NOT NULL,  
4   'password' varchar(128) NOT NULL,  
5   PRIMARY KEY ('username'),  
6   UNIQUE KEY 'email_UNIQUE' ('email'),  
7   UNIQUE KEY 'username_UNIQUE' ('username')  
8 );
```

```
1 CREATE TABLE 'AlbumContainsImage' (  
2   'album_id' int NOT NULL,  
3   'image_id' int NOT NULL,  
4   PRIMARY KEY ('album_id', 'image_id'),  
5   KEY 'id_image_idx' ('image_id'),  
6   CONSTRAINT 'id_album' FOREIGN KEY ('album_id') REFERENCES '  
7     Album' ('album_id') ON DELETE CASCADE ON UPDATE CASCADE,  
8   CONSTRAINT 'id_image' FOREIGN KEY ('image_id') REFERENCES '  
9     Image' ('image_id') ON DELETE CASCADE ON UPDATE CASCADE
```

```
8 );
```

```
1 CREATE TABLE 'Album' (  
2   'album_id' int NOT NULL AUTO_INCREMENT,  
3   'album_creator' varchar(32) NOT NULL,  
4   'album_title' varchar(64) NOT NULL,  
5   'album_date' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
6   PRIMARY KEY ('album_id'),  
7   KEY 'album_creator_idx' ('album_creator'),  
8   CONSTRAINT 'album_creator' FOREIGN KEY ('album_creator')  
9     REFERENCES 'User' ('username')  
10    ON DELETE CASCADE ON UPDATE CASCADE  
11 );
```

```
1 CREATE TABLE 'Image' (  
2   'image_id' int NOT NULL AUTO_INCREMENT,  
3   'image_uploader' varchar(32) NOT NULL,  
4   'image_title' varchar(64) NOT NULL,  
5   'image_date' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
6   'image_text' varchar(512) NOT NULL,  
7   'image_path' varchar(256) NOT NULL,  
8   PRIMARY KEY ('image_id'),  
9   UNIQUE KEY 'image_path_UNIQUE' ('image_path'),  
10  KEY 'image_uploader_idx' ('image_uploader'),  
11  CONSTRAINT 'image_uploader' FOREIGN KEY ('image_uploader')  
    REFERENCES 'User' ('username') ON DELETE CASCADE ON UPDATE  
    CASCADE  
12 );
```

```
1 CREATE TABLE 'Comment' (  
2   'comment_id' int NOT NULL AUTO_INCREMENT,  
3   'image_id' int NOT NULL,  
4   'comment_author' varchar(32) NOT NULL,  
5   'comment_text' varchar(512) NOT NULL,  
6   PRIMARY KEY ('comment_id'),  
7   KEY 'image_id_idx' ('image_id'),  
8   KEY 'comment_author_idx' ('comment_author'),  
9   CONSTRAINT 'comment_author' FOREIGN KEY ('comment_author')  
    REFERENCES 'User' ('username') ON DELETE CASCADE ON UPDATE  
    CASCADE,  
10  CONSTRAINT 'image_id' FOREIGN KEY ('image_id') REFERENCES '  
    Image' ('image_id') ON DELETE CASCADE ON UPDATE CASCADE  
11 );
```

## 2 Components

### 2.1 Model

- Album
- Comment
- Image
- User

### 2.2 Utils

The utils folder contains a set of support classes that simplify the development and maintenance of the application, without directly belonging to a single layer (Model, DAO, Controller, or View). Specifically:

- **DatabaseConnectionPool**: This class provides a pooling system for database connections, optimizing performance and managing the creation and release of connections.
- **DatabaseListener**: It initializes and destroys the connection pool at the startup or shutdown of the application (through the `contextInitialized` and `contextDestroyed` methods).
- **PasswordEncrypt**: It provides methods for password hashing and verification.
- **StringUtil**: It offers functions for string validation and manipulation (e.g., email, username, password, text verification, etc.), centralizing validation logic.
- **ViewEngine**: This class configures and manages the Thymeleaf templating engine.

### 2.3 DAOs

- **UserDAO**
  - `registerUser(user) : boolean`
  - `loginUser(user) : boolean`
  - `isUsernameTaken(username) : boolean`
  - `isEmailTaken(email) : boolean`
  - `getUsernameByEmail(email) : String`
  - `deleteUser(username) : boolean`
- **CommentDAO**
  - `getCommentsCountByUser(username) : int`
  - `getCommentsByImageId(imageId) : List of Comment`
  - `addComment(comment) : boolean`

- **AlbumDAO**

- `getMyAlbums(username)` : List of Album
- `getOtherAlbums(username)` : List of Album
- `createAlbum(album)` : boolean
- `getAlbumsCountByUser(username)` : int
- `getUserPersonalAlbumId(username)` : int
- `isAlbumOwnedByUser(albumId, username)` : boolean
- `getMyAlbumIds(username)` : List of int
- `getAlbumById(albumId)` : Album
- `doesAlbumExist(albumId)` : boolean
- `getImagesByAlbumId(albumId)` : List of Image
- `getImagesCountByAlbumId(albumId)` : int
- `getImagesByAlbumIdWithPagination(albumId, pageSize, startIndex)` : List of Image

- **ImageDAO**

- `getImagesCountByUser(username)` : int
- `addImage(image)` : int
- `updateImagePath(imageId, imagePath)` : boolean
- `addImageToAlbums(imageId, albumIds)` : boolean
- `doesImageExist(imageId)` : boolean
- `doesImageBelongToAlbum(imageId, albumId)` : boolean
- `getImageById(imageId)` : Image
- `doesImageBelongToUser(imageId, username)` : boolean
- `deleteImageById(imageId)` : boolean
- `getImagePathById(imageId)` : String

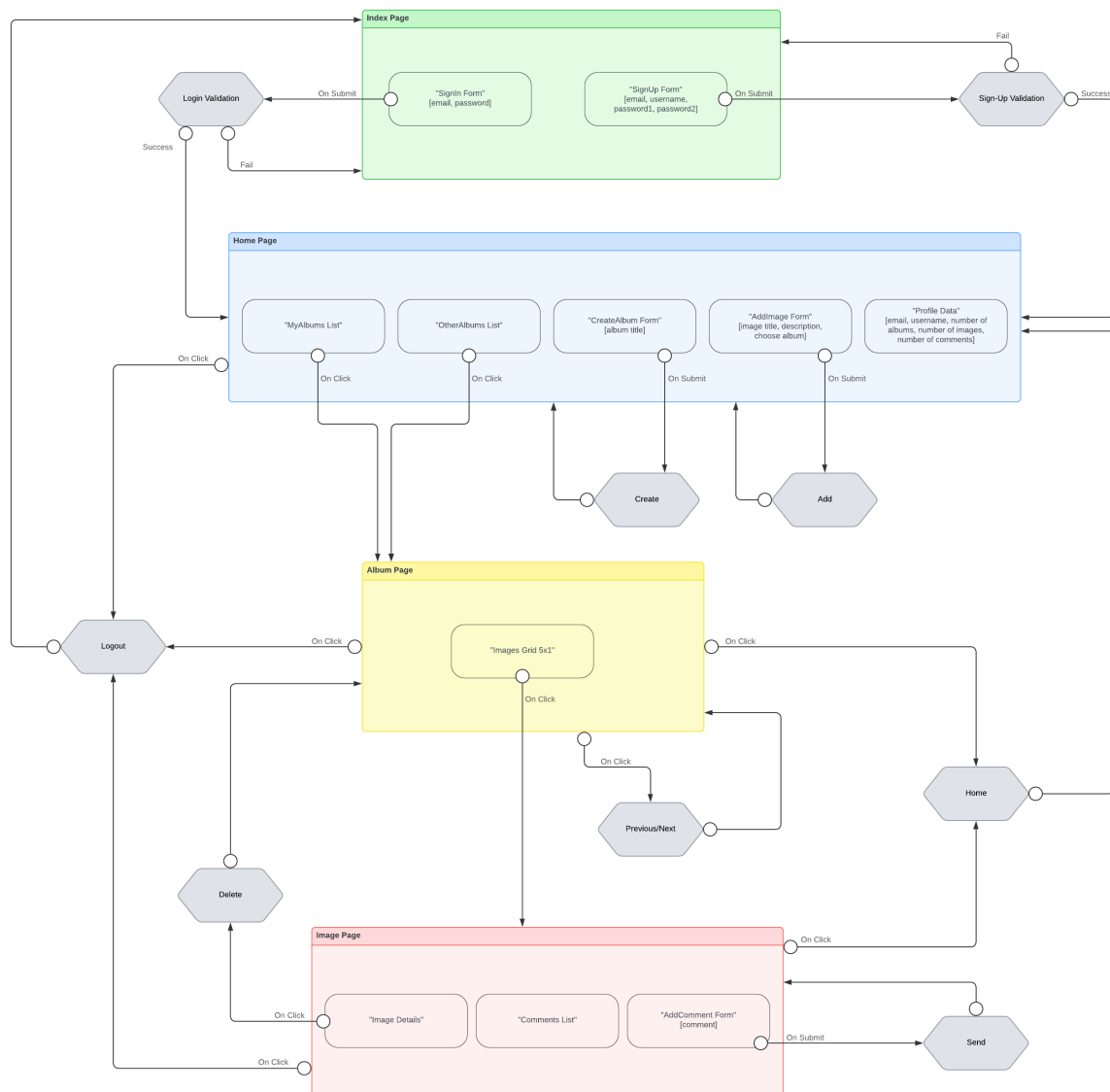
## 2.4 Controllers

- AlbumServlet
- ErrorServlet
- HomeServlet
- ImageServlet
- IndexServlet
- UploadsServlet

## 2.5 Views

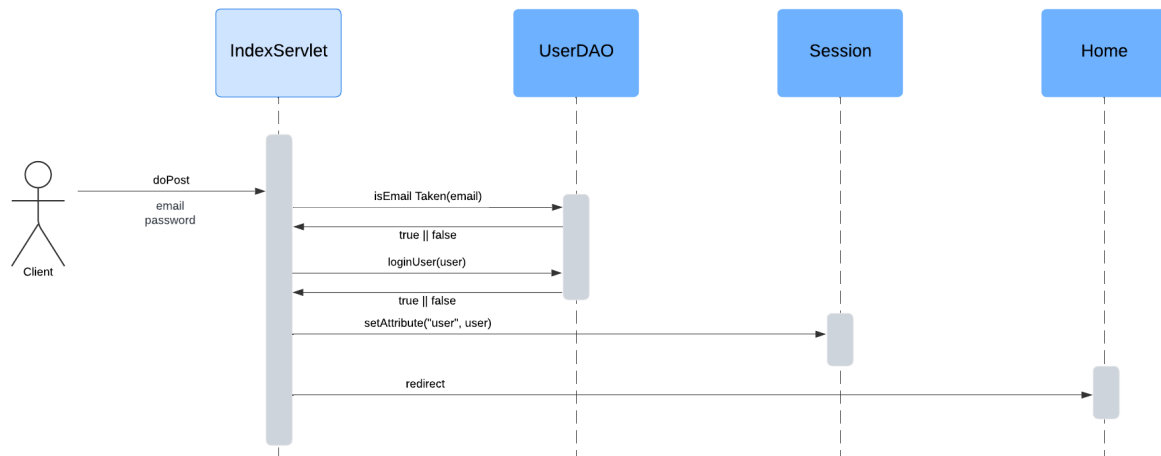
- album.html
- error.html
- home.html
- album.html
- image.html
- index.html

## 3 IFML Diagram

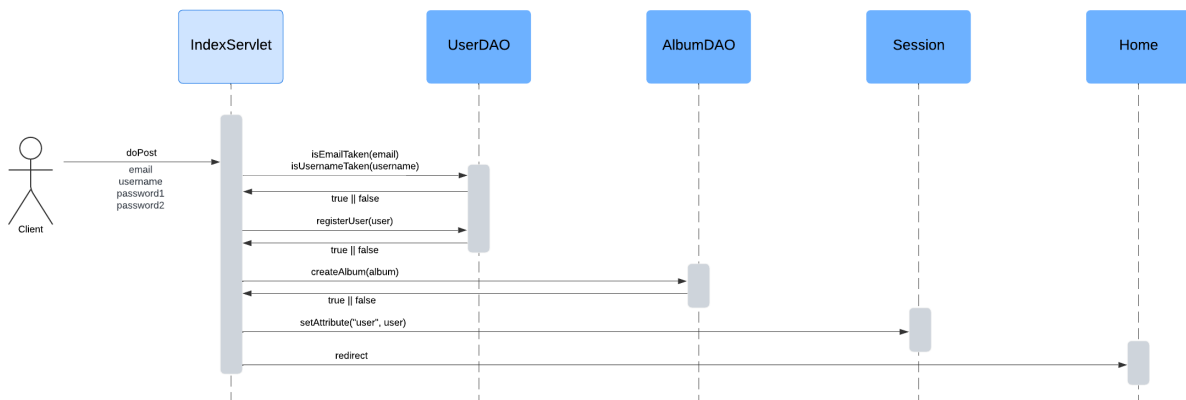


## 4 Sequence Diagram

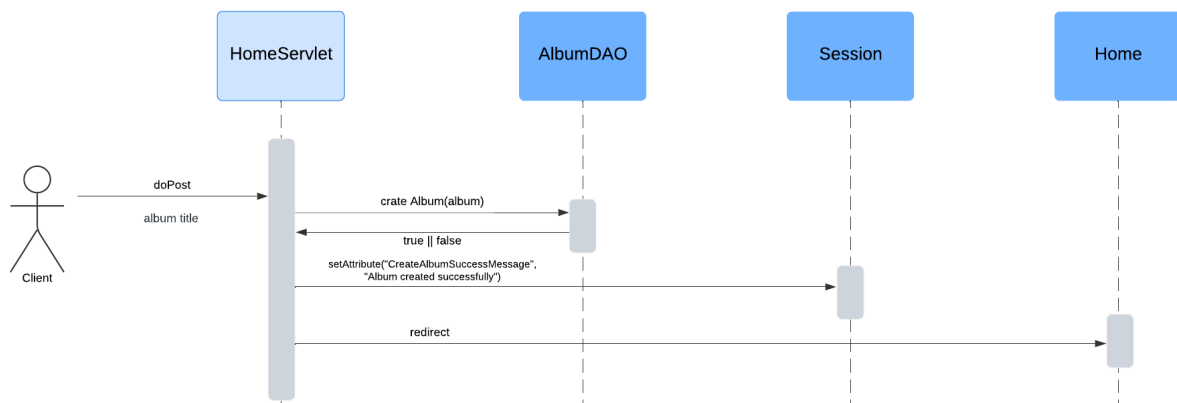
### 4.1 Event: Sign In



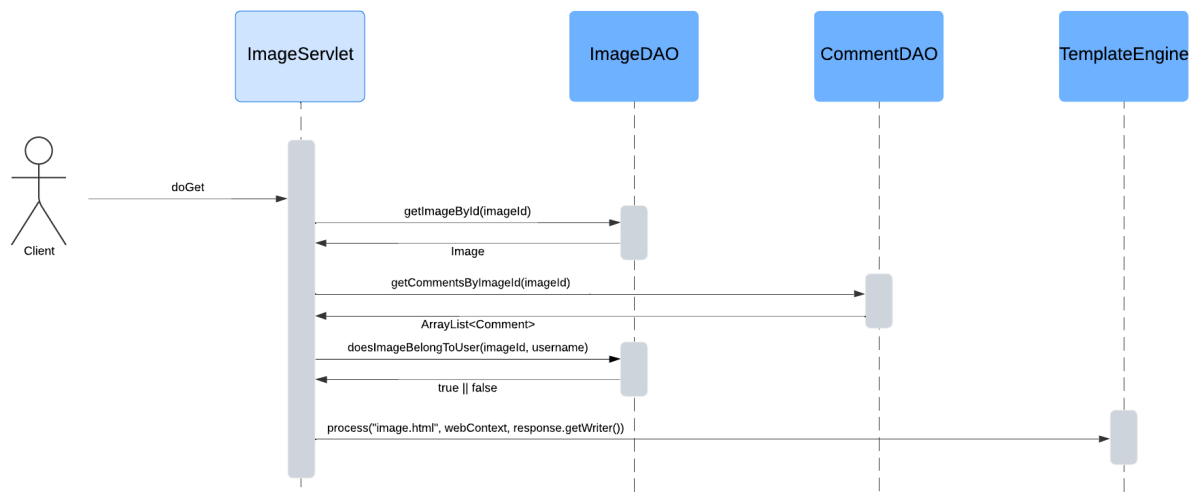
### 4.2 Event: Sign Up



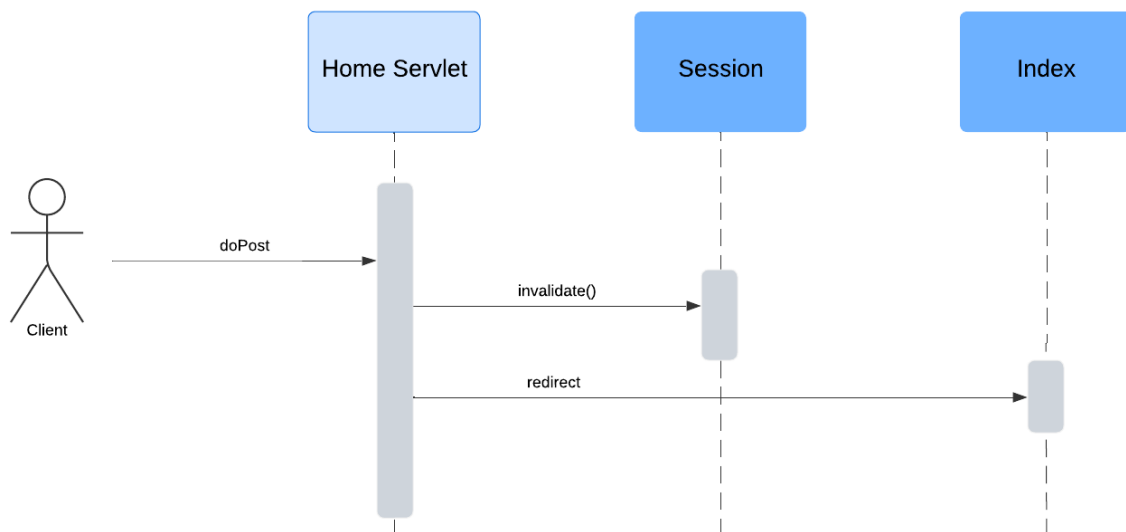
### 4.3 Event: Create Album



## 4.4 Event: Render Image Page

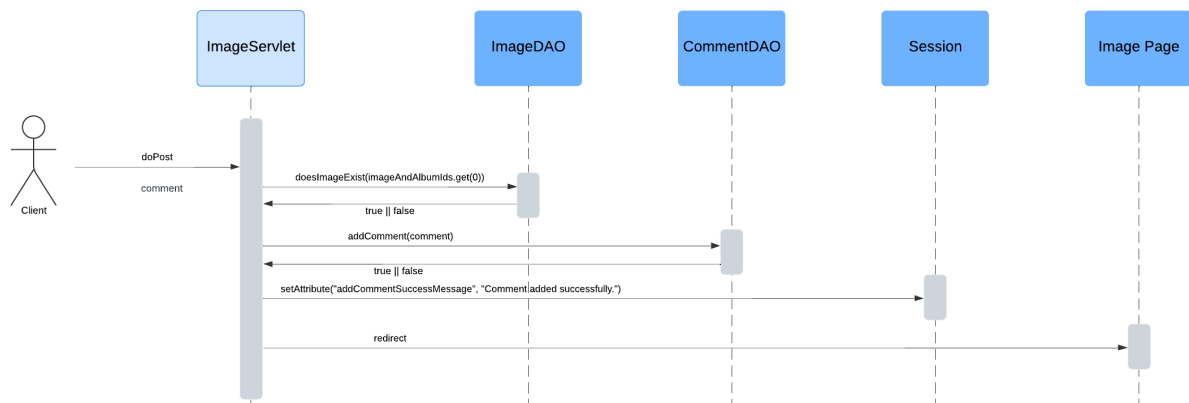


## 4.5 Event: Logout

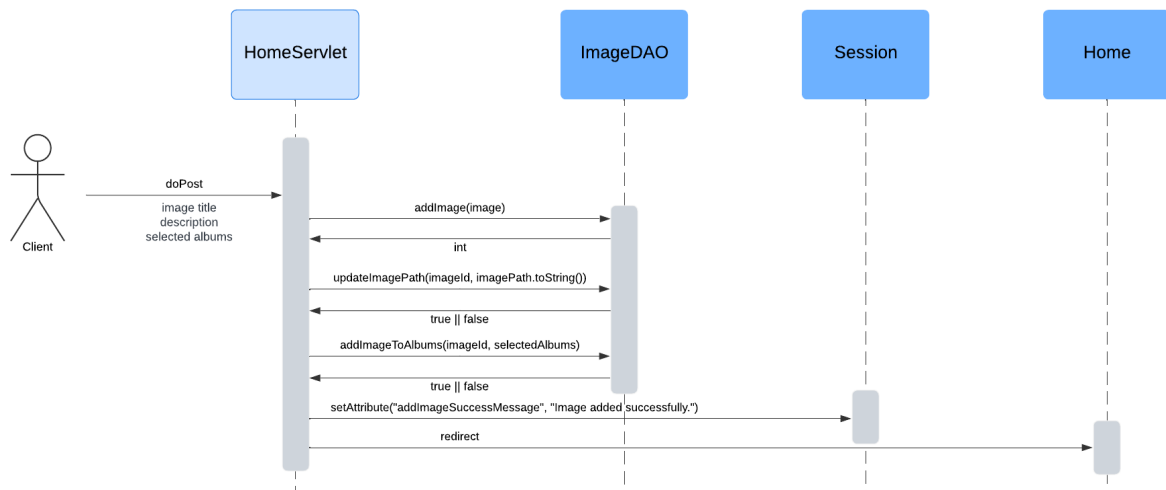




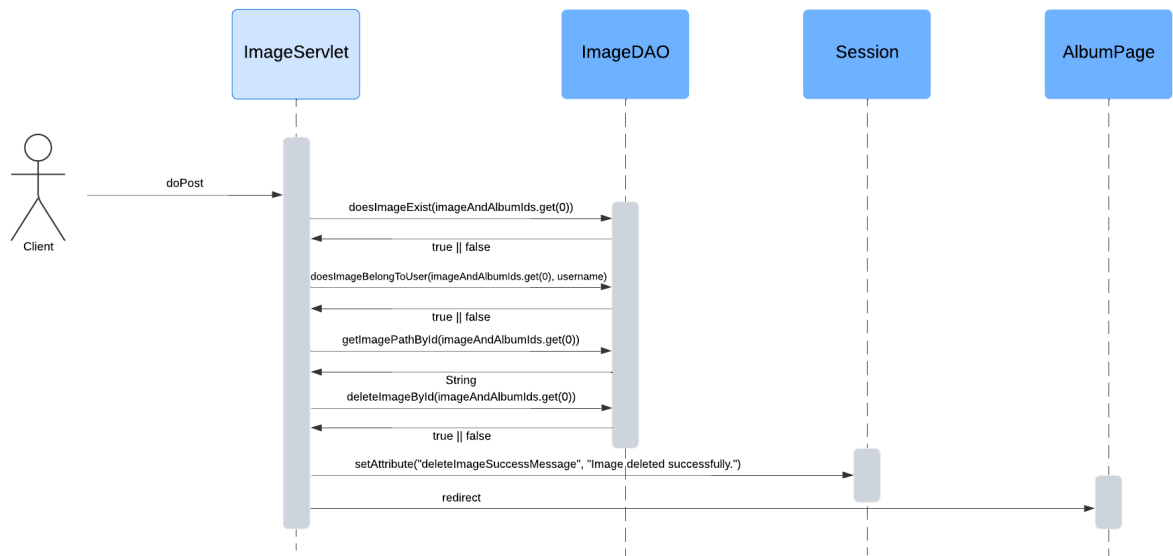
## 4.6 Event: Add Comment



## 4.7 Event: Add Image



## 4.8 Event: Delete Image



## 4.9 Event: Render Album Page

